

Assignment 6

Due Friday 20 October 2023, at the start of class

Please read Lectures 8,9,10,11,12 in the textbook *Numerical Linear Algebra* by Trefethen and Bau. This Assignment mostly covers Gram-Schmidt QR, Householder reflectors and QR, and least squares.

DO THE FOLLOWING EXERCISES from Lecture 8:

- **Exercise 8.3**

DO THE FOLLOWING EXERCISES from Lecture 9:

- **Exercise 9.3**

DO THE FOLLOWING EXERCISES from Lecture 10:

- **Exercise 10.2**
- **Exercise 10.3**

DO THE FOLLOWING ADDITIONAL EXERCISES.

P11. Either by using the built-in functions `polyfit()` and `polyval()`, or by setting-up linear systems and solving using Matlab's backslash command, reproduce Figures 11.1 and 11.2. Please make at least modest effort to duplicate the appearance of these Figures. (*Hints.* Note that `axis off` will generate a clean picture without unnecessary ticks and axes labels, and such. But then you might want to put back the axes themselves using `plot([-6 6], [0 0], 'k')` and similar commands.)

P12. The Matlab built-in `qr()` computes the QR factorization using Householder transformations as described in Lecture 10. In this problem, go ahead and use it. While we have used QR to solve linear systems, here we see that QR has a completely different purpose. For more, see Lectures 24–29.

(a) By searching for “unsolvable quintic polynomials” or similar, confirm that there is a theorem which shows that fifth and higher-degree polynomials cannot be solved using finitely-many operations including roots (“radicals”). In other words, there is no finite formula for the roots of such polynomials. Who proved this theorem and when? Show a quintic polynomial for which it is known that there is no finite formula. (*You do not need to prove your claim!*)

(b) At the Matlab command line, try the following:

```
>> A = randn(5,5);  A = A' * A;      % create a random 5x5 symmetric matrix
>> A0 = A;          % save a copy of the original A
>> [Q, R] = qr(A);  A = R * Q
...                  % repeat about 10 times
>> [Q, R] = qr(A);  A = R * Q
```

We start with a random, symmetric 5×5 matrix A_0 and then generate a sequence of new matrices A_i . Specifically, each matrix A_i is factored

$$A_i = Q_i R_i$$

and then the next matrix A_{i+1} is generated by multiplying-back in reversed order:

$$A_{i+1} = R_i Q_i.$$

Watch what happens to the matrix entries when you iterate at least 10 times. (*Use a for loop to see a strong effect from e.g. 100 iterations.*) What do you observe about this sequence of A_i ? Now compare `sort(diag(A))` to `sort(eig(A0))`.

(c) To see a bit more of what is going on in part **(b)**, show that

$$A_{i+1} = Q_i^* A_i Q_i.$$

This shows A_{i+1} has exactly the same eigenvalues as A_i ; explain.

(d) Write a few sentences which relate part **(a)** to what happens in parts **(b)** and **(c)**. (*Hint. Try to relate the two parts by yourself first. Then read Lecture 25 to confirm your understanding.*)